

UNITED STATES PATENT APPLICATION

FOR

**APPARATUS AND METHOD FOR SEPARATING CORRUPTED DATA
FROM NON-CORRUPTED DATA WITHIN A PACKET**

Inventor:

Sheng Li

PREPARED BY:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1026
(408) 720-8300

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number EL143570382US

Date of Deposit July 19, 2001

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231

LINDA BROST

(Typed or printed name of person mailing paper or fee)

Linda Brost 7/19/01
(Signature of person mailing paper or fee) Date

APPARATUS AND METHOD FOR SEPARATING CORRUPTED DATA FROM NON-CORRUPTED DATA WITHIN A PACKET

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0001] The present invention relates to transport protocols used in delivering real-time multimedia data such as audio and video data.

2. Art Background

[0002] The User Datagram Protocol (UDP) is a transport layer protocol commonly used in delivering real-time multimedia data such as audio and video data. Each data packet contains a UDP header and a payload. The UDP header contains a checksum field designed to protect the integrity of the entire data payload. Unfortunately, when an error is detected, the entire payload may be tossed out. If the data payload in the UDP packet contains multiple independent data segments, the undamaged segments will be tossed out as well. This either will increase network congestion by causing the sending device to resend the lost data, or will decrease the quality of the multimedia presentation since the lost data creates noticeable gaps in audio and/or video transmission.

SUMMARY OF THE INVENTION

[0003] A method and apparatus for detecting errors and improving quality in real-time data transmissions is provided. In one embodiment, the packet header checksum field is turned off to allow uninterrupted transmission of data packet payloads. A checksum added to each independent data segment in the datagram payload permits each data packet to be examined separately, resulting in improved transmission quality.

for the reason

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The objects, features and advantages of the present invention will be apparent from the following detailed description in which:

[0005] **FIG. 1** is a diagram of a User Datagram Protocol (UDP) header field common in the art.

[0006] **FIG. 2** is a diagram of a datagram provided by one embodiment of the present invention.

[0007] **FIG. 3** is a flowchart illustrating one method performed by one embodiment of the present invention.

[0008] **FIG. 4** is a flowchart illustrating another method performed by one embodiment of the present invention.

[0009] **FIG. 5** illustrates an apparatus according to one embodiment of the invention.

DETAILED DESCRIPTION

[0010] The User Datagram Protocol (“UDP”) is used in place of the transport layer of the network stack when reliable data delivery is not required. UDP is frequently used to transmit real-time audio and video where lost or corrupted packets are simply discarded (e.g., because there is no time to retransmit). UDP and TCP are both transport layer protocols.

[0011] UDP typically employs a checksum to ensure data integrity. As is known in the art, a checksum is generated by calculating the sum of the binary values in a block of data. The checksum is then transmitted with the underlying data. If the checksum indicates that a UDP packet (commonly referred to as a UDP “datagram”) contains corrupt data at the receiving end (e.g., at the client computer), the entire packet is simply discarded, with no further action being taken. Unfortunately, discarded packets may be comprised of a plurality of independent data segments, many of which may not be corrupt. As such, discarding an entire packet is inefficient way to deal with the problem of lost or corrupt data. Moreover, if the data is audio and/or video content, discarding multiple data segments in this manner may result in noticeably degraded audio/video playback at the client.

[0012] FIG. 1 is a diagram of a datagram 100 commonly used in the prior art. Datagram 100 consists of a header 101 and a data payload 102. Header 101 consists of a source port field 103, a destination port field 106, a length field 104 of the datagram 100, including header 101 and data payload 102, and a checksum 105. Data payload 102 consists of a plurality of independent data

segments. Source port field 103 is an optional field. When meaningful it indicates the port of the sending process, and may be assumed to be the port to which a reply should be addressed in the absence of any other information. If not used a value of zero is inserted. Destination Port Field 106 has a meaning within the context of a particular Internet destination address. Length Field 104 specifies the length in octets of the user datagram, including the header and the data. Checksum 105 is the 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the UDP header, and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets. The psuedo header conceptually prefixed to the UDP header contains the source address, the destination address, the protocol, and the UDP length. The checksum used for UDP packets is the same as used for TCP packets.

[0013] FIG. 2 illustrates a UDP datagram 200 employed in one embodiment of the invention. Datagram 200 consists of a header 201 and a payload 202. The header 201 may contain a source port field 209, a destination port field 211, and a length field 210. In this embodiment, the checksum field 209 has been set to zero, and a plurality of independent checksums 203, 205, and 207 have been calculated for a corresponding plurality of data segments 204, 206, and 208, respectively. Setting the UDP checksum field 209 to zero effectively turns off the checksum functionality, thereby ensuring that all data segments 204, 206, and 208 within the UDP packet are transmitted through the transport layer to the application layer or

other networking layer defined above the transport layer. At the application layer or other networking layer, the checksum for each data segment is independently validated, and only those individual data segments which are invalid are discarded.

[0014] Providing checksums for data segments encapsulated within transport layer data payloads provides a significantly more efficient way to transmit data, particularly when large packet sizes are defined at the transport layer (i.e., because more data segments may be encapsulated within larger transport packets). With respect to audio/video streams, this results in improved real-time streaming of audio and video data, because only corrupt data segments are discarded.

[0015] FIG. 3 illustrates a method according to one embodiment of the invention. At Block 301, the UDP header checksum is turned off by setting the checksum value to zero. At Block 302, a checksum is calculated and added to the data payload for each independent data segment. Independent data segments of various sizes and types may be used while still complying with the underlying principles of the invention. At Block 303, the entire data payload is sent to the application layer or other networking layer above the transport layer on the receiver side.

[0016] FIG. 4 illustrates a method according to another embodiment of the invention. At block 401, the data payload is received by the application layer (or other layer defined above the transport layer) on the receiving device. Error detection is performed at Block 402 by calculating checksums for each of

the independent data segments and comparing the calculated checksums to the checksums transmitted with the data segments. If the checksums match, then the data is presumed to be accurate. If, however, the checksums do not match, then the data is presumed to be corrupt. Independent data segments containing corrupt data are discarded at Block 403. Optionally, retransmission of these data segments may be requested at Block 404.

[0017] In one particular embodiment of the invention, GSM-AMR audio frames are individually assigned checksums and encapsulated within UDP datagrams. GSM is short for Global System for Mobile Communications, and is one of the leading digital cellular standards. If we choose a GSM-AMR rate of 4.75 Kbps as an example, each frame is 12 bytes long and includes 20 milliseconds of audio playback. Accordingly, if 30 GSM-AMR frames are encapsulated within a UDP packet, then the UDP packet will contain 360 bytes of data, not including the UDP packet header information. If a single burst error corrupts one of the 30 frames, the remaining 29 frames can be recovered using the techniques described herein. As such, the quality of the audio stream will be significantly improved.

[0018] As described above, embodiments of the invention may include various steps. The steps may be embodied in machine-executable instructions which may be used to cause a general-purpose or special-purpose processor to perform the steps. Alternatively, these steps may be performed by specific hardware components that contain hardwired logic for performing the steps, or

by any combination of programmed computer components and custom hardware components.

[0019] Elements of the present invention may also be provided as a computer program product which may include a machine-readable medium having stored thereon instructions which may be used to program a computer (or other electronic device) to perform a process. The machine-readable medium may include, but is not limited to, floppy diskettes, optical disks, CD-ROMs, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, magnet or optical cards, propagation media or other type of media/machine-readable medium suitable for storing electronic instructions. For example, the present invention may be downloaded as a computer program product, wherein the program may be transferred from a remote computer (e.g., a server) to a requesting computer (e.g., a client) by way of data signals embodied in a carrier wave or other propagation medium via a communication link (e.g., a modem or network connection).

[0020] One embodiment of an apparatus for processing data packets as described herein is illustrated in FIG. 5. A checksum calculation module 510 calculates checksums for each of a plurality of N data segments. A packet generation module 520 combines the calculated checksums with each of the plurality of data segments and incorporates the checksums and independent data segments within the payload of a data packet 540.

[0021] The characteristics of the data packet 540 generated by the packet generation module 520 depend on the specific packet generation

parameters 530 programmed into the module. For example, the parameters may define a packet of a particular size to be transmitted to a particular network address. As described above, in one embodiment, the parameters instruct the packet generation module 520 to set the checksum to zero (i.e., thereby effectively turning off the checksum if the data packet is a UDP packet). Various other packet generation parameters may be programmed consistent with the underlying principles of the invention.

[0022] A transmission module 550 then implements the remaining networking functions (e.g., at the network, data link and/or physical networking layers) required to deliver the data packet 540 to its destination. For example, if the data packet is a UDP packet, the transmission module may add an IP header to the data packet before sending the data packet over the network.

[0023] It should be noted that the various modules and functional parameters illustrated in FIG. 5 may be implemented in software, hardware, firmware or any combination thereof. For example, in one embodiment, the modules are configured in an application-specific integrated circuit ("ASIC"). In one particular embodiment, data packets are generated on a server as described above and transmitted to a personal computer or wireless device (e.g., a PDA, wireless phone, . . . etc).

[0024] Throughout this detailed description, for the purposes of explanation, numerous specific details were set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the system and method may be practiced without

some of these specific details. For example, while the embodiments described above focused on UDP, the underlying principles of the invention may be implemented in virtually any packet-switched environment in which packets are discarded. Moreover, while the technique for calculating data integrity described above is the "checksum," various other techniques for calculating data integrity may be employed.

[0025] In other instances, well known structures and functions were not described in detail in order to avoid obscuring the subject matter of the present invention. For example, it is presumed that one of ordinary skill in the art understands the basic principles of a layered network environment (e.g., the distinction between networking at the physical layer, the data link layer, the network layer, the transport layer, the application layer, . . . etc). A general discussion of these and other networking principles is set forth in ANDREW S. TANNENBAUM, COMPUTER NETWORKS (3rd Ed. 1996).

[0026] Accordingly, the scope and spirit of the invention should be judged in terms of the claims which follow.